



# Shadow Bug

## Shadow Bug Post-Mortem

Kim Valori & Juha Ylimäki  
Aalto University  
School of Arts, Design and Architecture  
Department of Media / Media Lab Helsinki

A thesis submitted for the degree of Master of Arts  
in New Media, Game Design and Production  
October 2017



---

**Author** Kim Valori, Juha Ylimäki

---

**Title of thesis** Shadow Bug Post-Mortem

---

**Department** Media

---

**Degree programme** MA in New Media: Game Design and Production

---

**Year** 2017

**Number of pages** 34

**Language** English

---

### **Abstract**

Shadow Bug is a platformer game developed as part of this thesis. The game differs from typical platformer games by combining the action of jumping with the action of attacking. Shadow Bug was first released on iOS and received critical acclaim in terms of positive press reviews. The game was featured by Apple on the App Store's front page upon release.

The written part of this thesis first presents an overview of Shadow Bug's features and achievements. Theoretical frameworks are then introduced for an analysis of the game's design. After presenting what Shadow Bug is, the thesis analyses the game's development process in terms of successes and failures. These successes and failures are presented and compared with experiences that other game development teams have had. Finally the lessons the development team learned during Shadow Bug's development process are reflected in terms of good game design practices and in terms of their effects on the future game development.

---

**Keywords** games, game design, platformer, post-mortem

---





---

**Tekijä** Kim Valori, Juha Ylimäki

---

**Työn nimi** Shadow Bug Post-Mortem

---

**Laitos** Media

---

**Koulutusohjelma** Taiteen maisteri, uusi media, pelisuunnittelu

---

**Vuosi** 2017

---

**Sivumäärä** 34

---

**Kieli** Englanti

---

### Tiivistelmä

Shadow Bug on tasoloikkapeli, joka kehitettiin osana tätä opinnäytettä. Peli eroaa tyypillisistä tasoloikkapeleistä yhdistämällä hyppytoiminnon hyökkäystoimintoon. Shadow Bug julkaistiin ensin iOS-alustalla ja peli sai hyvän kriittisen vastaanoton positiivisten lehdistöarvostelujen muodossa. Apple esitteli peliä App Store -sovelluskaupan etusivulla pelin julkaisun yhteydessä.

Tämän opinnäytteen kirjallinen osa esittelee aluksi yleisellä tasolla Shadow Bugin ominaisuuksia ja saavutuksia. Tämän jälkeen pelin suunnittelua analysoidaan teoreettisten viitekehyksien avulla. Kun Shadow Bug on pelinä esitelty, opinnäyte analysoi pelinkehittämisen prosessia onnistumisten ja epäonnistumisten näkökulmasta. Nämä onnistumiset ja epäonnistumiset esitellään ja niitä vertaillaan kokemuksiin, joita muut pelinkehitystiimit ovat kokeneet. Lopulta Shadow Bugin kehitystiimin oppimaa tarkastellaan hyvien pelisuunnittelukäytäntöjen ja tulevaisuuden pelikehityksen kannalta.

---

**Avainsanat** pelit, pelisuunnittelu, tasoloikka, jälkiselvittely

---

## Table of Contents

1. Introduction.....	5
1.1 Brief Description of the Game.....	5
1.2 Main Contributions.....	9
2. Division of Work.....	11
3. Theoretical Framework.....	13
3.1 Game Definition.....	13
3.2 Game Genres.....	14
3.3 Game Design Frameworks.....	15
4. Post-Mortem.....	17
4.1 What Went Right?.....	17
4.1.1 A Simple Unique Mechanic as a Strong Core.....	17
4.1.2 Focus on the Game Features.....	18
4.1.3 MVP Approach.....	19
4.1.4 Apple Featuring.....	19
4.1.5 The Benefits of a Small Team.....	20
4.1.6 Focused User Testing.....	20
4.1.7 Silhouette Art Style.....	20
4.1.8 Attending Game Conferences with an Agenda.....	21
4.1.9 Help from the Community.....	21
4.1.10 Right Choice of Tool: Unity.....	21
4.1.11 Pitching Competitions.....	22
4.2 What Went Wrong?.....	22
4.2.1 Unclear Role Division.....	22
4.2.2 Chemistry and Vision Problems.....	23
4.2.3 Solving Problems Others Have Already Solved.....	23
4.2.4 Steam Gamepad Control Scheme.....	24
4.2.5 Too Many Hats.....	25
4.2.6 Summer of Startups.....	25
4.2.7 Underestimating Development Time.....	26
4.2.8 Mobile Platform Not the Best for Premium Business Model.....	26
4.2.9 Planning Too Big.....	27
4.2.10 Silhouette Art Style.....	27
4.3 Post-Mortem Comparison.....	27
4.3.1 Similarities.....	28
4.3.2 Differences.....	29
4.3.3 Post-Mortem Comparison Results.....	30
5. Conclusion.....	31
5.1 Technical Lessons Learned.....	31
5.2 Graphics Lessons Learned.....	31
5.3 Post Shadow Bug Work.....	32
6. References.....	33



# 1. Introduction

Shadow Bug is a fast-paced puzzle-platformer game with a twist: jumping has been combined to attacking enemies. There are no separate jump and attack buttons in the game. The player character Shadow Bug leaps and attacks at the same time when the player clicks or taps on an enemy occupying the screen. This enables designing novel game puzzles and challenges, as the enemies acts both as hazards to avoid and means of navigation. Attacking with a leap is called jump attacking.

Shadow Bug started as a thesis project at Aalto University. The Shadow Bug team members were all eager to release their first game. In a couple of months the project started to grow into a more professional game project with commercial value and escalated into a company called Muro Studios. Apple featured the iOS version of Shadow Bug.

”Badland Meets Super Meat Boy” - TouchArcade (“Badland Meets Super Meat Boy’ Platformer ‘Shadow Bug’ Coming April 4th, New Trailer Released,” 2016)

”Shadow Bug is very well tuned to touch-screen controls.” - Hardcore Gamer (“Shadow Bug Speed-Slashes Onto iOS | Hardcore Gamer,” 2016)

”Shadow Bug is one of the most alluring games that has graced my device lately.” - AppAdvice (“Become one with the ways of the ninja in Shadow Bug,” 2016)

## 1.1 Brief Description of the Game

The title of the game is Shadow Bug and it has been released on Apple App Store, Google Play Store and Steam. The game is an action-adventure platformer with light puzzle-game elements. In Shadow Bug the player can move left and right, attack and leap through walls, collect orbs, push crates to different places, and push levers.

Shadow Bug is about controlling Shadow Bug’s moves in a 2D environment and exploring the game world. The gameplay is divided into three kinds of situations: attacking with a leap, avoiding obstacles and exploring the spaces to get forward. Attacking is mostly about fast reflexes and timing a path made of creatures in a right way in order to get to the next area in the level. Avoiding obstacles is about trying not to get caught by enemies, lasers, saws, or

spikes. Exploring is about searching secrets and the right path in the level and solving puzzles in order to advance in the level.

Shadow Bug is a black ninja bug with two katanas. Shadow Bug has probably practiced his whole life to be the ultimate ninja warrior in order to finally take the journey to take down the factory that pollutes Shadow Bug's world. In the Steam version Shadow Bug has a friend character, a fly that shows Shadow Bug where to attack.

The game world is occupied by many different creatures. The main opponents are Muros, marching creatures that are generated from a factory via strange pipes all around the forest. The world has many other creatures as well; flying moths, mad birds, spitting flowers, jumping fish, floating jellyfish, Muros with armor and laser guns, spiky one eyed boulders, fat Muros, floating drones, and 6 boss characters.

The story is set in a fantasy world where a huge factory is polluting the forest around it. A sewer connects the sealed factory and the forest. Shadow Bug eventually infiltrates the factory using this secret connection. The lesson in the story is that even a tiny character can make a difference with enough practise. The story of the game is very linear, but as the game is very speedrunning oriented, the levels in the game have many secret paths in order to play through the levels faster.



*The first level of Shadow Bug.*



*Shadow Bug finds the evil factory polluting the forest.*



*Shadow Bug attacks a flying monster.*

Shadow Bug's adventure takes the player through 36 hand crafted levels. The challenges vary from dodging spinning blades among treetops to solving puzzles in underground tunnels of toxic waste. The most novel parts of Shadow Bug are the 6 boss fights.





*The core of the factory.*

The world of Shadow Bug comes to life with layered parallax backgrounds. These colorful and picturesque landscapes are contrasted with the silhouettes of animated game characters. The pace of Shadow Bug's adventure is set by an epic soundtrack. The immersion of the game becomes complete with high quality sound effects.



*The sewers with pipes spawning monsters.*

## 1.2 Main Contributions

The greatest achievement for Shadow Bug is the featuring from Apple at the App Store front page. Shadow Bug also got numerous awards. The most notable are the Finnish Game Awards 2017 nomination, PAX East Indie Showcase selection 2016, Pocket Gamer Big Indie Pitch runner-up 2015, Summer of Startups Winning Pitch 2015 and Beatcon Best Pitch 2015.

The most positive press reviews are from AppAdvice, PocketGamer and Pelaaja. The Metacritic website combines all the press reviews together and counts the average score. The Metacritic score of Shadow Bug is 81 which is really good.

Here are 5-star reviews made by players at the App Store.

- “A simple, good & fun game.”

by Ealadon – Mar 23, 2017

TL;DR - it's fun, unless you're doing the challenges it isn't that hard. So I might have lied a little when I said it was simple. Often it is simple to beat the level, but if you want to earn three shurikens on each level you may find yourself playing any given level dozens of times on repeat. But that of course is why I love this game. It is not the easiest plat-former, but it also is not really challenging until you're combining the speed and score aspect. Because of that the "difficulty" of the specific challenge you set for yourself allows for a really nice curve. Finally, it is interesting to listen to (at first, of course all repetitive music gets old), and the large point globes that are hidden allow you to play through a level slowly and methodically to try to find where they are hidden. That is a wonderful mechanic to get people to notice all the little aspects that have went into making this game. If you like this sort of die, rinse, repeat games at all, you will like this one."

- “Addicting!”

by Eirejones – Jan 16, 2017

Super fun game. Difficult but not impossible. Lots of levels. Great graphics. Simple concept, outstanding execution.”

- “Best idea EVER!!!”

by The Reivewer – Nov 11, 2016

Who ever came up with this idea is a genius, Not only does everything look adorable, Its action packed and you feel like a beast when you fly through the air slicing bad guys to bits, and the simple but fast animation is perfect for this type of game. MURO studios... Keep up the good work!!! 😁😁😁😁😁😁”

Shadow Bug has an innovative core game mechanic which pushes the platformer genre forward. The core game mechanic serves well on mobile devices as you don't need any virtual joysticks or buttons on the screen to be able to move and attack in the game. The core game mechanic also works great on desktop with mouse and keyboard.



## 2. Division of Work

This is a joint Master's thesis of two authors who were part of creating Shadow Bug as a team, in accordance with the guidelines of Media Lab Helsinki. The authors are Kim Valori and Juha Ylimäki. The development team behind Shadow Bug had a third member as well: Veli Laamanen. Laamanen's main contribution was the designing and producing audio for the game including the soundtrack as well as the sound effects. Laamanen also contributed to many other design aspects of the game, the largest of them being level design.

Kim Valori is the programmer of the development team. He is also the game's lead designer and thus contributed on all of the game's design aspects. Shadow Bug is based on a prototype game created by Valori. Valori is the author of the following chapters of this thesis:

### 2. Division of Work

#### 3.1 Game Definition

#### 3.3 Game Design Frameworks

### 4. Post-Mortem (Chapter Overview)

#### 4.1.1 A Simple Unique Mechanic as a Strong Core

#### 4.1.5 The Benefits of a Small Team

#### 4.1.10 Right Choice of Tool: Unity

#### 4.2.1 Unclear Role Division

#### 4.2.3 Solving Problems Others Had Already Solved

#### 4.2.4 Steam Gamepad Control Scheme

### 5.1 Technical Lessons Learned

### 5.3 Post Shadow Bug Work

Juha Ylimäki is the artist of the development team and created all of the game's graphics and animations. Like the other team members, Ylimäki also contributed on many design aspects of the game. Ylimäki is the author of the following chapters of this thesis:

### 1.1 Brief Description of the Game

### 1.2 Main Contributions

### 3.2 Game Genres

#### 4.1.2 Focus on the Game Features

#### 4.1.3 MVP Approach

- 4.1.4 Apple Featuring
- 4.1.6 Focused User Testing
- 4.1.7 Silhouette Art Style
- 4.1.8 Attending Game Conferences with an Agenda
- 4.1.9 Help from the Community
- 4.1.11 Pitching Competitions
- 4.2.2 Chemistry and Vision Problems
- 4.2.5 Too Many Hats
- 4.2.6 Summer of Startups
- 4.2.7 Underestimating Development Time
- 4.2.8 Mobile Platform Not the Best for Premium Business Model
- 4.2.9 Planning too Big
- 4.2.10 Silhouette Art Style
- 4.3 Post-Mortem Comparison
  - 4.3.1 Similarities
  - 4.3.2 Differences
  - 4.3.3 Post-Mortem Comparison Results
- 5.2 Graphic Lessons Learned

### 3. Theoretical Framework

#### 3.1 Game Definition

Greg Costikyan defines a game as "an interactive structure of endogenous meaning that requires players to struggle toward a goal" (Costikyan, 2002). This definition can be used to analyze different components of a game to make comparison with other games easier. Costikyan's definition has five elements that can be used to analyze specific components of a game: interaction, structure, endogenous meaning, struggle and goals.

In Shadow Bug's case, endogenous meaning and goals are similar to many other platformer games. With endogenous meaning Costikyan refers to elements that have meaning only inside a game. Therefore Shadow Bug's typical level elements such as monsters, platforms and collectables fall under endogenous meaning just like in Nintendo's Super Mario Brothers and Team Meat's Super Meat Boy for example. The goals in these games are very similar as well: the player needs to finish all levels in the game. Shadow Bug offers additional goals in forms of improving one's level-finish times and discovering collectables as well, just like Super Meat Boy.

Shadow Bug's biggest difference to other platformer games is in the way the player controls the game: the game's interaction. The main interaction of Shadow Bug is tapping monsters to attack them. This is different from other platformer games in which the core dynamic is often built with traditional jumping in which the player presses a button on a controller or a touch screen and this makes the game's protagonist jump in the air.

The novel core interaction of Shadow Bug is the foundation of the game's structure; the composition that the game's rules and elements form for the player to play in. Since Shadow Bug's rules allow the player to defeat monsters easily and move fast and somewhat freely, playing the game is focused on timing and dodging instead of grinding and traditional combat. Parts of Shadow Bug's structure resemble other platformer games with powerful movement mechanics such as Super Meat Boy and Moon Studios' Ori and the Blind Forest. In both of these games, just like in Shadow Bug, the player is using more time in avoiding dangerous elements in the environment with mechanics of movement instead of fighting monsters.

The last element of Costikyan's game definition is struggle, which means the challenges the player needs to face to reach the game's goals. Shadow Bug has two kinds of challenges: a small



amount of puzzles that require the use of wits, and a great amount of mechanical scenarios that require timing and mastering the controls of the game. While puzzles and mechanical scenarios are typical for platformer games, the specifics of these challenges are novel in Shadow Bug because of the core interaction.

### 3.2 Game Genres

Games can be categorized into different genres based on their characteristics. Miikka Junnila introduces various genres in his thesis “Genret ja niiden rikkominen pelisuunnittelussa 2007” (Junnila, 2007). Junnila argues that there are genres based on the game mechanics of the game and genres based on the story of the game. Junnila lists game mechanics genres as Action, Racing, First Person Shooter, Action-Adventure, Platformer, Strategy, Real-Time-Strategy, Management, Sports, Role-Playing, Online Role-Playing, Puzzle, Adventure, Music, and Simulator. Junnila lists story genres as Adventure, Arcade, War, Fantasy, Sports, Sci-fi, Crime, Realism, Horror, and Abstract.

In the game mechanic genres Shadow Bug falls into Action-Adventure because it includes a linear structure, a story, and puzzle solving in order to advance in the game. Shadow Bug is also a Platformer as the player walks and jumps in a 2D world filled with platforms.

In the story genres Shadow Bug falls into Fantasy because the game world is a fantasy world filled with strange creatures and the world doesn't represent the real world in any way. Shadow Bug is also a Sci-Fi as the world includes a factory with weird technology, machines and laser beams. Shadow Bug has quite a lot of different themes in its story so it can also be categorized into the Arcade genre.

According to Junnila a game is based on its game mechanic genre and the story genre that merge together creating the game world with all the components in the game. These components are the characters, objects and resources in the game. All these components are a mixture of a certain game mechanic and a part of the story. With these components the player can take actions in the game world via the user interface created for the game.

The core game mechanic in Shadow Bug is the ability of the player character to attack and leap to targets and create chain attacks by leaping from one target to another just by clicking or tapping on the creatures occupying the screen. Shadow Bug is a unique Platformer game as the player doesn't have a separate jump and attack buttons. The core game mechanic is novel in the Action-Adventure Platformer genres as it explores a single unique mechanic and how to get the most out of it. The theme of a black shadow ninja warrior fits this core game mechanic as most

players can understand the logic behind ninja leaping huge distances slicing creatures at a single blow.

The closest related games in game mechanic genre are Super Meat Boy, Hollow Knight, Ori and the Blind Forest, Rayman and Owlboy. The games mentioned are much larger than Shadow Bug but they hold the same Action-Adventure Platformer mentality. The closest related games in story genre are Badland and Limbo because of the black silhouette visual style.

### **3.3 Game Design Frameworks**

It can be useful to analyze a game using a framework such as MDA (Hunicke, LeBlanc, & Zubek, 2004) or Kaleidoscope of Effective Gamification (Kappen & Nacke, 2013). MDA can be used to analyze a game's rules and systems, and the kinds of sensations they invoke in the player. Kaleidoscope of Effective Gamification on the other hand focuses more on the different sources of motivation that the game offers the player. Since Shadow Bug's design goals included invoking specific feelings in the player, MDA is a good framework for analyzing the game's different components.

The acronym MDA comes from the words mechanics, dynamics and aesthetics. These are the three levels of abstraction that MDA uses to compartmentalize a game's design. Mechanics are the building blocks of the game: the different actions and rules that dictate what kind of events are possible in a game. When put together, these mechanics create systems with purpose, like puzzle and battle scenarios. These systems are a game's dynamics.

Like in most platformer games, in Shadow Bug the player can make the protagonist walk left and right. This is an example of a mechanic. The main mechanic of Shadow Bug is the jump attack. The player can click on a monster to perform a jump attack on it. This will cause the protagonist to move to the monster in a straight line and then defeat the monster. Another mechanic is having the protagonist die upon touching a laser beam and another one is that a laser beam can move up and down. With these mechanics, dynamics can be built. For example, if we take a moving laser beam and place a monster behind it, a scenario is created in which the player has to time a jump attack correctly or they will die from hitting the laser beam. Timing attacks correctly is a very common dynamic in Shadow Bug.

In terms of MDA, aesthetics are different types of pleasures the player derives from a game. MDA categorizes pleasures in eight groups: sensation, fantasy, narrative, challenge, fellowship, discovery, expression, and submission. Four of these are central pleasures that Shadow Bug intends to provide the player: sensation, challenge, discovery, and fantasy.

Fantasy is a very typical pleasure that digital games produce for the player. In terms of MDA, fantasy means a fictional setting that the player can immerse themselves in. In Shadow Bug's case, this setting is the game's story, world and characters. All of this is built for the pleasure of fantasy: to allow the player to escape reality by immersing themselves in the world of Shadow Bug.

The pleasure of sensation means sensory pleasure. The visuals and sounds mentioned above provide this kind of pleasure, but the intended main source for sensory pleasure in Shadow Bug is its core mechanic of jump attacking. Since jump attacks are the action that the player performs the most, it was important to make performing them feel good. Fluid controls are one of the main design focuses on Shadow Bug and the way it feels to control the character is designed to provide the player great pleasure of sensation.

Based on feedback and design goals Shadow Bug is not an easy game. Shadow Bug is intended to provide the player the pleasure of challenge. Finishing a level in Shadow Bug is a challenge of its own, but Shadow Bug was designed to have replayability value by rewarding the player when finishing levels without using much time and when collecting a level-specific amount of orbs. The player is awarded up to three shurikens after completing the level based on their performance. In addition to this, each level has a global leaderboard in which players can compare their level-finish-times to those of other players.

The final central pleasure Shadow Bug is designed to provide the player is the pleasure of discovery. Each level of Shadow Bug has a hidden giant orb for the player to find. Finding the orbs is rewarded by animations and a mark on the game's map on each level in which a giant orb has been found. In addition, some levels have hidden shortcuts that can be used to complete the levels faster. Finding these secret orbs and shortcuts gives the player the pleasure of discovery.

## **4. Post-Mortem**

This chapter presents aspects of Shadow Bug's development that were particularly successful followed by ones that were particularly unsuccessful. Finally, this chapter presents a comparison of our experiences with those of other developers' presented in an analysis by Michael Washburn Jr. et al. (Washburn, Sathiyarayanan, Nagappan, Zimmermann, & Bird, 2016).

### **4.1 What Went Right?**

#### **4.1.1 A Simple Unique Mechanic as a Strong Core**

A lot of platformer games have jumping as a central action that the game's dynamic is built on. In Shadow Bug we didn't include a traditional jump mechanic but instead created the jump attack mechanic. We had not seen this mechanic used in other games so using it allowed us to build something new.

Having a simple mechanic like the jump attack was a powerful tool for designing Shadow Bug. We based all of the game's design on the jump attack mechanic. In particular the design of the game's levels and the elements in them stemmed directly from the core mechanic. For example, the enemies in Shadow Bug had to be utilized differently than in most games. Having a core mechanic that allows the player to defeat enemies by merely tapping them caused them to be less threatening. In Shadow Bug the enemies are mainly a tool for the player for move from one place to another instead of being a threat.

Having a unique core mechanic allowed us to think of new ways in which traditional platformer game enemies could be utilized. The best example of this is a wall climber, an enemy that is seen in a lot of platformer games, such as Nintendo's Metroid. A wall climber is an enemy that moves along the game world's platform surfaces, whether they be floors, walls or ceilings. Shadow Bug's mechanic allowed us to design scenarios where wall climbers can reach areas that the player can't and so the player has to wait for the wall climbers to reach a specific point before attacking them in order to advance. We were also able to build scenarios in which the player has to protect the wall climbers from other enemies so that they could reach the area that the player is trying to get to. Having a unique core mechanic didn't only enable us to create something new; it also enabled us to use old ideas in new ways.

When designing scenarios for Shadow Bug, we approached the design process with a mentality that the scenario should not be possible in any other game. This reinforced having the jump attack as the seed of all design as it is the unique mechanic that separates Shadow Bug from other games. A good example where this design mentality can be seen is Core Guard, one of the boss battles of Shadow Bug. Core Guard is a boss battle without platforms, which is rare in platformer games. The player is constantly forced to use the jump attack mechanic in this boss battle to prevent them from falling into a deadly laser beam. At the same time the player has to dodge the boss enemy's attacks and try to attack the enemy itself once an opening arises in order to advance. Designing boss battles was easier once the game's mechanic was taken as a starting point for the design, because the mechanic limited options of the battle narrowing the design to a certain scope but at the same time served as a source of inspiration.

The jump attack mechanic did not only help in designing the mechanical aspects of Shadow Bug. It also made it easier to design the game's story and aesthetics. For example, the protagonist of the game is a super hero ninja, because that role fits the idea of performing jump attacks and being able to defeat enemies easily. The ninja theme can not be only seen in the game but also heard, as the soundtrack has oriental influences.

#### **4.1.2 Focus on the Game Features**

We were planning a lot of different features to add depth to the game but fortunately we were able to realise what features were really needed in order to keep the focus and get the game published eventually. We got a lot of good insight about what we should focus on when we tested the game with people who had never played the game before. Best cases were when some professional designers happened to play our game in some events where we were showcasing the game and they immediately noticed some important flaws in our focus with certain features we had implemented in the game.

The best example is the shop feature we had been working on at least for few weeks until we heard from several people that we should definitely cut it out as it will take a lot of time from us and it doesn't really give value to the game's core design. After a few days we thought the shop feature through and understood that it makes sense to cut it out as it saves time from huge amount of coding, art asset creation and user testing and gives us more time on focusing on what makes the game stand out, which is the level design that takes the core game mechanic to the next level and makes the game unique.



### **4.1.3 MVP Approach**

When we were planning a feature we wanted to add to the game the first thing we did was of course trying to figure out in our heads how it would affect the game, what possible problems it might have and would it be worth implementing. After an hour of talking and planning there always comes a moment when you should realise that you can not figure out in your head how the feature will feel and play out in the game. So the fastest possible way to figure out if something works or not is to create an MVP (minimum viable product) version of the feature and test if it makes any sense at all. This way you will save time from all the thinking about could it work or should it be done. Just do it and try it out. We saved a lot of time this way.

We also used the same MVP approach when we were implementing features that will stay in the game. Meaning when we were happy enough about some feature we said to ourselves it's done and didn't use anymore time on that particular feature. Unless we found the extra time in the end of the project when we started polishing everything in the game.

### **4.1.4 Apple Featuring**

The greatest achievement for Shadow Bug is the featuring from Apple at the App Store front page. We learned from the Finnish game community that as a small indie team making a mobile title without any marketing budget, the only credible marketing strategy is finding an Apple contact face to face in some event and doing certain things to keep Apple happy about your game in order to have better chances in getting the featuring. Some of these things include sending a roadmap of your game to your Apple contact and updating it when necessary, promising a content update or two after the release of your game for Apple, using Apple's technology in your game, and creating a UI that is animated smoothly and is pleasant for the player to use.

Apple also featured Shadow Bug for one week as the free app of the week which gave us millions of downloads which is great when thinking about the future of the company. Having more people knowing Shadow Bug is a great base for creating more.

#### **4.1.5 The Benefits of a Small Team**

Having a small team had great advantages. Communication worked well and it was easy to be democratic. Having a small team also cut costs in a few ways. First of all there was no need for an office, since with only three people it wasn't hard to find places to work in. We worked at several different locations: at our homes, at Aalto University, at cafes and at an accelerator work space. With a small team we didn't need a lot of money for salaries either, so our budget remained small. A small budget meant that we didn't need to make a large profit to stay afloat as a company.

Another benefit of having a small team was that we could change our strategy quickly. We didn't suffer from the bureaucracy of a large company so if we needed to change direction, we could do it fast. An example of this is the shop system we were going to implement in the game. After working on the shop system for a month we tested it with players and as it was not working well, we realized that we would have to scrap it entirely. We made the decision immediately as we saw that players didn't understand how to use the system and its purpose in the game and started developing other features the next day.

#### **4.1.6 Focused User Testing**

As mentioned earlier we were testing the game a lot with people who had never tried the game before and got valuable feedback from almost every session we had. The biggest iterative process on a single feature was with the tutorial that is in the beginning of the game. We knew that it's the most important thing to get right so we were testing each iteration with people who had never tried the tutorial before to get new feedback on the current version. This took a lot of time but without the feedback Shadow Bug would have a disastrous tutorial.

#### **4.1.7 Silhouette Art Style**

The art style of Shadow Bug was a clever choice because it saved a lot of time from the creation of graphic assets and level decoration. The black silhouette assets were really fast to draw. The animations were faster to produce as we didn't have to worry how certain parts look on top of each other during the animation, all that mattered was the silhouette of the character being animated. Also the decoration of the levels was fast because of the same reason. All we had to worry about was how the silhouette looks, not about how all the assets look on top of each other.

#### **4.1.8 Attending Game Conferences with an Agenda**

We learned quite early in our journey that when you are at an event without a certain agenda, then you will most likely not get so much out of it. Sure you can mingle around, and you should, but it's always less effective if you don't know what you are after. So we attended most of the conferences with a certain mission in mind. For example; finding an Apple contact, finding a publisher, finding an investor, and getting feedback from players about some features in the game. It always helps you to focus if you know what you are doing at the event. Also if you aren't a seasoned veteran in visiting game conferences you should know that if you go to the event with your friends, then you will most likely not meet as many people as you would if you were there alone. It's always easier to go where your friends are going in the event and so you miss many opportunities in meeting new people.

#### **4.1.9 Help from the Community**

One of the biggest assets we have had is the developer community in Helsinki. When we started the project as students we had no idea what we were getting into and we had no idea how we would do most of the stuff we needed to do during the project. Fortunately we have the community of very experienced game developers around us that we could ask almost anything. In most cases we got an answer to our question and if not they always introduced us to someone who knew something about the topic at hand. We got a couple of developers even meeting us and showing us how to do a very specific feature in the game. We would have never made Shadow Bug as successful as it is without the help of the community. If you are ever in doubt about something, remember to ask for help.

#### **4.1.10 Right Choice of Tool: Unity**

We chose to develop Shadow Bug using Unity, because it is a tool we were all familiar with from our previous projects. For us, it was critical to use a tool that removed the need for creating several engines, such as a rendering engine and a physics engine ourselves. With a small team like ours, creating such engines would have been implausible because of the time it would have consumed, not to mention that creating engines is not something our team knew how to do. Unity also saved us a lot of time with its ability to test a game fast without building it.

Unity's popularity was helpful as well: since Unity has so many users, there are a lot of tutorials and other helpful resources available on the Internet. In addition, it was easy to find other developers who use Unity, not to mention Unity being used by our teachers in Aalto

University. Most of the time if we didn't know how to solve a problem in Unity, it was easy to find someone who did.

Porting the game to different platforms such as Google Play Store and Steam would have been impossible for us without a tool like Unity. The work we had to do regarding porting the game was mostly integration regarding a platform's leaderboard and achievement system and in the case of Steam creating PC-compatible control schemes. In other words, we didn't have to work on making the game run on other platforms at all, since Unity automated that for us.

#### **4.1.11 Pitching Competitions**

The best part about pitching your game is not really the competition or the award you might win in the pitching competition. The most useful part is that you really have to think what your game is about. If you practise the pitching many times you eventually start to get a clearer image of what you are doing with your project. Of course the awards you win in the pitching competitions always gives your project some credibility which helps in sales and selling your project to possible partners and investors.

We were quite a good team when it comes to practicing for pitching competitions. Most of the times we got feedback from each other on the text part of the pitch and after training the pitch alone we always pitched to each other as many times as it was necessary in order to be able to pitch smoothly when the actual pitching competition happened.

### **4.2 What Went Wrong?**

#### **4.2.1 Unclear Role Division**

For some part the division of roles in Shadow Bug's development was very clear from the beginning. Being a programmer, Valori would produce code, being an artist, Ylimäki would produce graphics, and being a sound designer, Laamanen would produce sound. However, game development consists of aspects not listed earlier as well, and the responsibilities of those aspects should have been clearly defined early in the development.

There were many areas that we all participated in, although it would have been more effective to delegate one area for one person. An example of this is marketing: we agreed that everyone

would contribute as there was no team member who wanted to take responsibility of this specific area. This meant that all of us had to learn social media practices and other marketing methods, while it would have been enough for only one of us to do that. Another example is level design, in which everyone wanted to participate. This caused the level design to have some inconsistency in it in terms of quality and continuity which in turn lead to more iteration and polishing with the levels than necessary. The lack of delegation in development areas cost a lot of time.

Another problem was not having a vision holder, as all the team members had slightly differing visions of the final product. This caused lost time in terms of unnecessary conversations of the game's vision and team members developing the game to different directions.

#### **4.2.2 Chemistry and Vision Problems**

From the very beginning we felt that chemistry between all of us did not work as it should have worked. We all thought that our desire to create something more ambitious was bigger than some small chemistry problems but we were wrong. The longer we continued with the team the bigger the chemistry problems grew. This took a lot of energy from all of us but still we somehow managed to make the game and publish it.

We all had different vision in our heads how the final game should be like. This is very normal, but since we had the chemistry problems, we couldn't figure out who's vision we would take as the leading one early enough to save a lot of time from many nonsense discussion sessions. If we didn't have to use so much time on these problems then we would have been able to create a better game and save ourselves from many sleepless nights.

#### **4.2.3 Solving Problems Others Have Already Solved**

Although Unity was a great choice for an environment to develop the game in, we didn't leverage it as much as we could have. One of Unity's benefits is Unity Asset Store, where one can purchase various assets made by other developers to be used in one's own Unity projects. These assets include all kinds of resources one would use in game development, such as 3D models, textures and scripts. We started leveraging the Asset Store many months into development when we needed to access native features of various platforms, but we should have done so earlier. After the positive experiences with Asset Store we realized that we could have saved a lot of time by using other people's solutions to problems that we were solving ourselves.



#### 4.2.4 Steam Gamepad Control Scheme

When porting Shadow Bug to PC and more specifically Steam, we researched some of the aspects that could affect the game's commercial success on the platform. For example, we looked at the features some of the successful platformer games on Steam had. All of the games we researched such as Inside, Hollow Knight and Owlboy had support for a gamepad controller. We decided that Shadow Bug would have that too. This turned out to be a mistake for two reasons: our core game mechanic is not very compatible with a gamepad and we greatly underestimated the amount of time implementing this feature would take.

From the beginning we knew that controlling the game with a keyboard and a mouse would work better than controlling it with a gamepad. Still, we felt that having the option of using a gamepad would be attractive to players as it would enable playing the game from the comfort of one's couch for example. However, having gamepad support listed as a feature in Shadow Bug's Steam store page meant that players who have a gamepad would not even try the keyboard and mouse controls before trying the game with a gamepad. After all, most platformer games work better with a gamepad than keyboard and mouse and so players would assume that this is true with Shadow Bug as well. We received feedback saying that the gamepad controls are clunky and that using a keyboard and a mouse works better. It is possible that we should have implemented a disclaimer in the game or its store page saying that gamepad is recommended for couch gaming only and that the use of a keyboard and a mouse is recommended.

We assumed implementing the gamepad controls would take approximately two weeks, when in reality it took seven. We thought the biggest problem was creating fluid controls for the core gameplay itself, but we ended up using most of the gamepad implementation time in creating navigation schemes for the user interface menus of the game. This could have been foreseen with better planning. Another aspect we failed to take into consideration early in the implementation was that different gamepads behave differently and require implementation regarding their compatibility with the game. We ended up integrating third party software to handle this aspect, but did it so late that we had to refactor a lot of our code to make the integration work. In hindsight, the amount of time we used implementing the gamepad controls should have been used for other tasks. That said, having support for gamepad had unintended benefits as well. Some players reported enjoying the extra challenge that playing the game with a gamepad provided compared to playing with a keyboard and a mouse.

#### **4.2.5 Too Many Hats**

The blessing and the curse of being a small indie team is that you have to do all by yourself. Meaning that you have the power to decide what you want to do as a company and that there are numerous tasks consuming time from the development. Handling different type of paper stuff, paying salaries, signing contracts, figuring out tax numbers to USA for being able to sell games on Steam. Of course you have an accountant for doing the most of the taxes and accounting, but still you have to give them the material and keep your paper logistics clear to avoid any problems from the tax office. All this paperwork is not overwhelming or difficult but it takes a lot of time from all the stuff you actually would like to be doing.

So if possible try to have one dedicated person who is willing to do all the paperwork. A business developer who goes to all the events to meet people and making new contacts and facing the players of your game and pitching for possible investors or publishers. A community manager who takes care of your players and fans in different forums and social media. We have been doing all these things at the same time while we have been developing our game. It has been a huge learning process to understand a bit of all aspects of game development business but it has also taken a huge cut from the quality and features of our game.

#### **4.2.6 Summer of Startups**

Summer of Startups is an incubator for startups in Otaniemi during summer. We got in and we were happy about it. We learned different aspects of startup life and the culture but not really so much new from game development or game business. The whole summer we were at the Summer of Startups was great but regarding our game project it didn't really help at all. We learned new things and met new people in the startup scene which is great but at the same time we lost a lot of good development time and again this cost us on the features and quality of the game.

The best part about Summer of Startups was that we met people at the startup scene and we have got some opportunities to participate in different events, for example in China, for free because people in the right place happened to know us from Summer of Startups.

#### **4.2.7 Underestimating Development Time**

When we started creating Shadow Bug we were planning to make a small mobile game with a few different worlds and enemies and bosses and you can see where this is going. This has been a big learning process for us as well. Development takes a lot more time than you first think. This is true in most cases. We were planning a lot of new features for the game but fortunately we met many people at different game events telling us that we should focus more on the core mechanic of the game in order to save time and be able to publish the game one day.

#### **4.2.8 Mobile Platform Not the Best for Premium Business Model**

We made the game for mobile first because we thought that the core game mechanic is perfect for mobile. We were right in some ways. We got featured by Apple which is a huge thing giving us some time to figure out what to do next. When thinking about the core audience, who are the players we make the game for, we made a wrong decision on going for mobile. Shadow Bug's mechanic is great for mobile devices but the game we wanted to create was for core gamers, people who play with PC and consoles. Also the game is a premium game meaning that the game has a price the player has to pay before he can play it. This means the game we wanted to create would have fit better on desktop and consoles first. In Shadow Bug the current difficulty curve is also designed for more mobile audience because we went for mobile first. This makes the game less appealing for PC audience.

When going for mobile you have to create a freemium game with ads and in-app-purchases. Also the game should be playable with one finger tapping in portrait mode instead of horizontal. This gives you a much bigger audience on mobile. So with these things considered we can see Shadow Bug as a premium PC game fitted on a mobile device just because it happens to have a mechanic that works smoothly on a mobile device. But if the audience is wrong, then the fact that the game works perfectly on mobile does not really give you that success you are after with your game. Also when going for mobile you have to be careful that your game works on most of the mobile devices, meaning that you have to optimize a lot. We would have had more time for interesting graphic assets and game features if we would have made the game for PC because we wouldn't have had to consider all the restrictions on mobile.

We were lucky with the Apple feature. It gave us about ten thousand sold copies of Shadow Bug iOS version. We got the Free App of the Week feature from Apple as well later that year and Shadow Bug got 2 million free copies downloaded with it. If we would have developed Shadow Bug with PC and console audience in mind we wouldn't have got this much visibility with the current version of Shadow Bug. If we developed Shadow Bug with PC and console

audience as our first customers, then we should have made the game larger with more content and features.

#### **4.2.9 Planning Too Big**

At the beginning of the development of Shadow Bug we were planning a lot of features that we would have never had the time to implement. The most time consuming would have been the story and all the graphic assets, extra animations, camera movements and different features needed to code to make all the story elements come together in a right way. Fortunately we figured out on some point of the development that we will go with gameplay first. This meant that all the story elements and cutscenes were secondary when it came to prioritising our tasks.

Before we figured this out we had many long discussions about the story and how it should be shown to the player and what all kind of amazing environmental animated assets we will create to make the story really come alive. All these story elements would have been great to see in the game but if we would have invested our time in creating those elements, then the gameplay and the game itself would have suffered a lot.

#### **4.2.10 Silhouette Art Style**

While the black silhouette art style has been a great time saver for us, at the same time it might have cost us the opportunity to get the most players for the game because it doesn't look original enough. In some cases people might think it has the same game mechanic as Badland because it looks similar. Badland is a great and successful game that everyone knows so it's probably good and bad to have a game that looks similar. If we would have got more original visual style on Shadow Bug we would have got more players because people wouldn't have had any prejudices on how the game works just by seeing the screenshots of the game. The screenshots and video at the App Store are very important as they give most of the possible customers their first impression how the game is played. If they think they have already played something similar, then they will most likely not buy the game.

### **4.3 Post-Mortem Comparison**

Washburn analyses 155 different post-mortems from different development teams in his article '“What Went Right and What Went Wrong”: An Analysis of 155 Postmortems from Game Development' (Washburn et al., 2016). In the article Washburn shows how 22 different

aspects of game projects failed and succeeded. We compare our failures and successes with the Washburn's 22 different aspects of game projects with the 155 post-mortems he analysed.

The 22 different aspects of game projects that Washburn lists are; Art, Creativity, Features, Game Design, Gameplay, Product Evolution, Scope, Development Process, Documentation, Obstacles, Team, Testing, Tools, Budget, Hardware, Publisher Relations, Schedule, Community Support, Feedback, Marketing, Piracy / Licensing, and Other. Other includes: teams believing in themselves, a team self funding their own project, luck, and risks paying off.

### **4.3.1 Similarities**

These aspects of the Shadow Bug project were a success as well as in most of the cases in Washburn's article:

Creativity: We were able to discuss about our ideas openly within the team to keep the creativity flowing.

Features: We managed to keep the main focus of the game on the core game mechanic and the level design around it.

Game Design: The core game mechanic of Shadow Bug is unique and it kept the team motivated.

Gameplay: Shadow Bug is developed with gameplay first in mind. Therefore the gameplay is smooth and doesn't have many flaws or bugs.

Scope: We managed to keep the scope of the game small enough to be able to make the end product as polished as it is.

Development Process: We managed to make quick prototypes of our planned features and tested them with new players and ditched the poor features quickly.

Testing: We managed to test our game in many different game events and locations. We also got successfully a couple of hundred beta testers who gave us important feedback on Shadow Bug.

Tools: We programmed our own tools to make different processes inside Unity faster when it was necessary.

Community Support: We have got a huge amount of support from our community in testing and giving feedback on the game during it's development.

Feedback: We asked and received a lot of feedback which we used in order to make the game as good as we possibly could.

Other: We were lucky that we always got fruitful feedback on the most critical design and feature decisions from our testers. The team members also believed in themselves a lot and



everyone had the dream of publishing their first own game. These helped the project to get done. We were also lucky that Apple featured Shadow Bug on the App Store.

These aspects of Shadow Bug project were a failure as well as in most of the cases in Washburn's article:

Documentation: As a small development team there was no sense in using time on documenting our processes and designs. If some new person should make something out of our designs or processes without our help then that would be really hard as there are no documents on anything.

Schedule: Our schedule stretched a lot from what we originally planned. This is because everything in the process of game development from start to publishing and creating content updates were new for us. Also the founding and running of the company and all contracts were new for us and they took a lot of time.

#### **4.3.2 Differences**

Art and Team aspects of the Shadow Bug project were a success and a failure, whereas in most of the cases in Washburn's article they were a success:

Art: Players always give good feedback on the visuals of Shadow Bug and the black silhouette style was fast to create. The black silhouette style is not anything new so the game doesn't stand out as much as it could with more unique art style.

Team: Everyone in the team was very motivated and talented in their own profession. However the chemistry didn't work with everyone as it should have which complicated our communication and slowed down our processes. Also because we were new to everything our role casting was poor in the beginning which hindered the development.

Obstacles in the Shadow Bug project is a success, whereas in most of the cases in Washburn's article it was a failure. We had obstacles with the team chemistry but we managed to overcome them and work on our tasks properly even though we were slowed down by the weak chemistry.

Marketing in the Shadow Bug project is a failure, whereas in most of the cases in Washburn's article it was a success. Shadow Bug was a hit on iOS because of the featuring Apple gave us, not because we were good at marketing. The Steam release was not so good for Shadow Bug because we were not good at marketing the Steam release. We also never had a budget to do any professional marketing. We tried some Facebook marketing with a small amount of money but we didn't know how to get the most out of it. On Twitter we managed to get many new followers before the Steam release but it was not enough to consider it a success.

### **4.3.3 Post-Mortem Comparison Results**

The Shadow Bug project was not so different in its successes and failures when reflected with the 155 post-mortems analysed by Washburn. The biggest difference seems to be when other teams have encountered an obstacle they failed handling it properly in most cases. In our project the biggest obstacle was the weak chemistry between some of the team members which we overcame with communication, passion for the product and hard work ethics in order to get the game finished, polished and published.

## **5. Conclusion**

### **5.1 Technical Lessons Learned**

When starting a new project in the future, we will proceed with the initial prototyping in a very similar fashion as with Shadow Bug: we will develop most of the required features ourselves. But after validating the concept with a simple prototype, we will attempt to utilize Unity's Asset Store and other similar resources earlier in our next projects. The potential amount of time and effort saved by researching and testing solutions others have made seems to be too great to pass up on.

Shadow Bug was designed without much regard for commercial aspects. We chose to publish the game on mobile devices first and are now left to wonder what would have happened if we would have targeted another platform such as PC first. We didn't consider where the potential audience for the game's genre is, but instead chose the first platform based on the game's mechanics. When making our next game, we will put more effort into choosing a platform.

### **5.2 Graphics Lessons Learned**

Choosing the black silhouette art style for Shadow Bug was a great time saver but also a possible mistake when considering the sales of the game. When considering the amount of work on all the foreground graphic assets including the characters that needed animation I think the black silhouette style did save us a lot of time which then again can be seen in the quality of game testing and other aspects of the game development process. This considered, for our next project we will try to find a more original visual style in order to stand out from the crowd because we aim to achieve a commercially successful game project whereas Shadow Bug started as a student project.

Our next project will most likely be developed for PC and consoles first, which means we have a much more computing power in our use. This gives us the possibility to create more vivid game world with more detailed graphic assets. A great advantage would be if we manage to find a person who knows how to work with certain types of shaders. This would give us more workspace on the visual style of our next game project. In Shadow Bug we found some unorthodox ways in Unity to represent some certain visual aspects of the game. Figuring these out was fun, but if we would have had a professional on the task we would have got a better looking game.

### 5.3 Post Shadow Bug Work

Shadow Bug Rush is a game we developed after Shadow Bug using some of the same assets as in Shadow Bug. Shadow Bug Rush uses the same mechanics of controlling the protagonist, but differs a lot in terms of the game's overall dynamic. Where in Shadow Bug the player attempts to finish all the levels of the game, Shadow Bug Rush's levels can't be finished. Shadow Bug Rush has two levels with partly procedurally generated content and the player's task is to defeat as many monsters as they can before dying.

Shadow Bug Rush's monetization model was inspired by advertisement based games such as Hipster Whale's Crossy Road. The development time before releasing Shadow Bug Rush was only three months compared to Shadow Bug's eighteen. Shadow Bug Rush has generated slightly more revenue than Shadow Bug, and has shown to be a good way to leverage a game's assets after release instead of just completely moving on to new projects. Like Shadow Bug, Shadow Bug Rush was also featured by Apple on App Store's front page.

We are planning to create a bigger game than Shadow Bug using the jump attack mechanic. We hope to find new layers of depth to the mechanic by combining it with other mechanics. Creating Shadow Bug left us at a unique position with knowledge and experience of using the jump attack mechanic and we feel we should leverage that. The game is most likely going to involve some aspects of the metroidvania genre meaning a large interconnected world to explore.

We feel that progress in the design of 2D platformer games has been very slow over the last few decades and that the improvements in the genre have been mostly technical. We hope that we can inspire game developers to try new mechanics in old genres or even use our jump attack mechanic in new ways.

## 6. References

- “Badland Meets Super Meat Boy” Platformer ‘Shadow Bug’ Coming April 4th, New Trailer Released. (2016, March 22). Retrieved August 29, 2017, from <http://toucharcade.com/2016/03/22/shadow-bug-release-date/>
- Become one with the ways of the ninja in Shadow Bug. (2016). Retrieved August 29, 2017, from <https://appadvice.com/review/shadow-bug>
- Costikyan, G. (2002). I Have No Words & I Must Design: Toward a Critical Vocabulary for Games. Retrieved from <http://www.digra.org/wp-content/uploads/digital-library/05164.51146.pdf>
- Hunicke, R., LeBlanc, M., & Zubek, R. (2004). MDA: A Formal Approach to Game Design and Game Research. *ResearchGate*, 1. Retrieved from [https://www.researchgate.net/publication/228884866\\_MDA\\_A\\_Formal\\_Approach\\_to\\_Game\\_Design\\_and\\_Game\\_Research](https://www.researchgate.net/publication/228884866_MDA_A_Formal_Approach_to_Game_Design_and_Game_Research)
- Junnila, M. (2007). Genret ja niiden rikkominen pelisuunnittelussa. Retrieved from <https://aaltodoc.aalto.fi:443/handle/123456789/12484>
- Kappen, D. L., & Nacke, L. E. (2013). The Kaleidoscope of Effective Gamification: Deconstructing Gamification in Business Applications. In *ResearchGate*. <https://doi.org/10.1145/2583008.2583029>
- Shadow Bug Speed-Slashes Onto iOS | Hardcore Gamer. (2016). Retrieved August 29, 2017, from <http://www.hardcoregamer.com/2016/04/04/shadow-bug-speed-slashes-onto-ios/199472/>
- Washburn, M. W., Sathiyarayanan, P., Nagappan, M., Zimmermann, T., & Bird, C. (2016). “What Went Right and What Went Wrong”: An Analysis of 155 Postmortems from Game Development. *Microsoft Research*. Retrieved from

<https://www.microsoft.com/en-us/research/publication/what-went-right-and-what-went-wrong-an-analysis-of-155-postmortems-from-game-development/>